

---

---

## – ДОПОЛНИТЕЛЬНОЕ ОБРАЗОВАНИЕ –

---

УДК 371.398:62

### ГЕЙМИФИКАЦИЯ КАК СПОСОБ ФОРМИРОВАНИЯ ИНЖЕНЕРНОГО МЫШЛЕНИЯ

*Белолубский Михаил Михайлович,  
старший педагог дополнительного образования  
МБУ ДО «Центр технического творчества»  
городской округ «город Якутск»*

**Аннотация.** В статье рассматриваются особенности внедрения элементов геймификации в процесс обучения программированию в учреждениях дополнительного и общего образования. Обозначаются преимущества игрового подхода, повышающие мотивацию, вовлеченность и самостоятельность учащихся, а также создающие условия для формирования устойчивого интереса к изучению языков программирования. На основе анализа практического опыта и цифровых образовательных платформ приводятся примеры применения геймификации и рекомендации по разработке учебных материалов.

**Ключевые слова:** геймификация, программирование, цифровые технологии, обучение, мотивация, игровые механики, дополнительное образование.

### GAMIFICATION AS A WAY TO FORM ENGINEERING THINKING

*Belolyubsky Mikhail Mikhailovich,  
Extracurricular education Senior teacher  
Technical Creativity Center  
Municipal budgetary institution of additional education  
The city of Yakutsk municipal district*

**Abstract.** The article discusses the features of the introduction of gamification elements in the process of learning programming in institutions of additional and general education. It outlines the advantages of the game approach, increasing the motivation, involvement and independence of students, as well as creating conditions for the formation of a sustained interest in learning programming languages. The article gives the examples of the use of gamification and recommendations for the development of educational materials based on the analysis of practical experience and digital educational platforms.

**Keywords:** gamification, programming, digital technologies, motivation, game mechanics, extracurricular education.

#### **Введение**

Современное образование претерпевает существенные изменения под влиянием процессов цифровизации. Особенно заметны эти изменения в тех областях, где обучение требует высокого уровня абстракции, логического мышления и способности

самостоятельно анализировать сложные задачи. Одной из таких областей является программирование, которое становится актуальным не только в старших классах, но и в начальном и среднем школьном звене, а также в системе дополнительного образования.

Несмотря на широкую распространённость курсов по программированию, многие учащиеся испытывают трудности на начальных этапах. По данным педагогических наблюдений, значительная часть детей сталкивается с проблемами мотивации уже после нескольких занятий, когда обнаруживает, что программирование требует терпения, усидчивости и системности [1].

В этих условиях геймификация становится эффективным инструментом, позволяющим сформировать эмоционально позитивную образовательную среду и повысить интерес к освоению программирования. В отличие от традиционных подходов, геймификация не заменяет учебный процесс, а делает его более осмысленным, наглядным и эмоционально насыщенным.

### **Теоретические основы геймификации**

Под геймификацией понимается использование игровых механизмов в неигровой учебной среде. Её ключевое отличие от учебных игр заключается в том, что геймификация не создаёт отдельную игру – она «встраивает» игровые элементы в уже существующий образовательный процесс [1].

К элементам геймификации относятся:

- уровни и индивидуальные траектории,
- прогрессия сложности,
- баллы и рейтинг,
- достижения, награды, виртуальные значки,
- система миссий и испытаний,
- сюжетные задания,
- внутренняя валюта или ресурсы,
- обмен опытом между участниками.

Психологи отмечают, что игровая модель соответствует особенностям детского и подросткового мышления – ученику важно видеть результат своих действий сразу, отслеживать собственный рост и иметь возможность вернуться к решению задачи, если с первого раза оно не было найдено [1,2]. В этом смысле геймификация создает интерактивную модель обучения с постоянной обратной связью, чего зачастую не хватает традиционному уроку.

### **Опыт исследования. Геймификация в обучении программированию**

Программирование обладает важной особенностью: успех ученика напрямую связан с количеством практических попыток. Чем больше ошибок ученик совершает и исправляет, тем быстрее он осваивает алгоритмическое мышление [3]. Однако это же свойство делает предмет сложным – многие дети испытывают психологический дискомфорт из-за ошибок, воспринимая их как «неудачу».

Геймификация решает эту проблему. Ошибка в игровой среде воспринимается как «шаг назад», но не как катастрофа: ученик может попробовать снова, но уже с новым опытом.

Можно выделить несколько преимуществ игрового подхода:

#### **Повышение мотивации**

Система баллов, достижений, переходов на новые уровни и внутриигровых поощрений помогают ученику:

- видеть собственный прогресс,
- сравнивать свои результаты с предыдущими,
- стремиться к новым целям.

#### **Формирование внутренней заинтересованности**

Ребёнок не только выполняет задание, но и:

- получает удовольствие от процесса,
- действует самостоятельно,
- стремится завершить начатую работу.

#### **Развитие алгоритмического мышления**

Игровая среда:

- предоставляет разнообразные задания,
- позволяет повторять попытки без последствий,
- усиливает понимание причинно-следственных связей.

Ученики начинают естественно рассуждать логически, потому что видят результаты своих решений «здесь и сейчас».

#### **Создание коммуникативной среды**

Большинство современных платформ позволяют:

- обмениваться проектами,
- давать комментарии,
- участвовать в общих челленджах.

Это укрепляет учебное взаимодействие и формирует культуру сотрудничества.

### **Примеры платформ и инструментов**

#### **Code.org**

Предлагает пошаговые задания с элементами игры, где каждая задача – уровень с требуемым действием. Ученик получает мгновенную обратную связь и награды, что повышает динамику обучения.

#### **Scratch**

Позволяет создавать собственные мини-игры и анимации из логических блоков. Учащиеся видят результат своих действий сразу, что повышает интерес и к экспериментированию, и к программированию.

#### **CodeCombat**

Пользователь выполняет задания в формате квестов, управляя персонажем с помощью написанного кода. Это создаёт плавное погружение в Python или JavaScript.

#### **Roblox Studio**

Эта среда подходит для более старших обучающихся. Здесь они создают полноценные игровые миры, учатся проектировать логику, взаимодействие объектов, экономику и поведение персонажей.

#### **Kahoot!**

Сервис позволяет проводить тестирование и повторение материала в форме соревнования. Такие занятия повышают эмоциональный настрой и удерживают внимание.

## Алгоритм разработки учебных материалов с элементами геймификации

### 1. Постановка цели

Прежде чем интегрировать игровые механики, педагог должен определить:

- чему научится ученик,
- какие компетенции будут сформированы,
- как измерить результат.

Игра не должна затмевать обучающую задачу.

### 2. Продуманная структура уровней

- каждый уровень сложнее предыдущего,
- новое задание требует освоить определённую тему,
- ученик может вернуться и перепройти уровень.

Это создаёт ощущение естественного роста.

### 3. Система поощрений

Она должна быть понятной и прозрачной.

- медали за упорство,
- баллы за креативность,
- значки за решение задач без подсказок.

### 4. Обратная связь

- регулярная,
- персонализированная,
- позволяет понять, что ученик сделал хорошо, а что нужно улучшить.

### 5. Тестирование и доработка

- пробный запуск на небольшой группе,
- сбор отзывов,
- корректировка заданий.

Геймификация – гибкая система, требующая постоянной настройки [1].

## Примеры учебных заданий

В рамках обучения программированию с использованием элементов геймификации педагог может применять разнообразные формы деятельности, направленные на развитие алгоритмического мышления и самостоятельности учащихся. Наиболее эффективными являются:

- **соревнования по отладке кода**, в ходе которых учащиеся ищут и исправляют ошибки в предложенных программах на время;
- **квестовые задания на прохождение уровней**, где каждый новый этап требует применения изученных команд и логических конструкций;
- **мини-проекты**, например, создание простых интерактивных заданий или алгоритмов на платформе Code.org;
- **групповые разработки**, когда учащиеся распределяют роли: руководитель проекта, исполнитель алгоритмов, тестировщик, дизайнер интерфейса;
- **портфолио достижений**, включающее успешно пройденные уровни, полученные награды, выполненные проекты и индивидуальные улучшения.

Подобные формы работы позволяют учащимся видеть собственный прогресс, сравнивать результаты, осознавать достижения и испытывать эмоциональную удов-

летворённость от процесса обучения. Это значительно повышает уверенность в своих силах и формирует устойчивую учебную мотивацию.

### **Практический пример фрагмента занятия**

В рамках курса «Основы программирования» в МБУ ДО «ЦТТ» города Якутска был проведён практический модуль «Создание игры “Крестики-нолики” на Python». Занятие было построено с использованием игрового подхода: прохождение этапов, система достижений за правильно реализованные функции и визуальная фиксация прогресса.

Ход занятия:

1. Этап 1 – Создание игрового поля:

Учащиеся написали функцию, формирующую двумерный список 3×3, который отображал текущее состояние игрового поля. За успешную реализацию выдавалось достижение «Стартовый разработчик».

2. Этап 2 – Обработка хода игрока:

Дети реализовали ввод координат клетки и проверку корректности хода. Условия на занятии были представлены в форме мини-квестов:

- «Защити поле от неправильного ввода»,
- «Нельзя ставить две фигуры в одну клетку».

3. Этап 3 – Проверка победы:

Это был самый сложный этап. Учащиеся должны были создать функцию, которая определяет победную комбинацию по строкам, столбцам и диагоналям. За выполнение ученики получали уровень «Алгоритмист».

4. Этап 4 – Итоговая игра:

После соединения всех функций дети запускали полноценную игру «Крестики-нолики» на Python. На этом этапе многие улучшали игру самостоятельно: добавляли визуальное оформление в консоли, вводили повторную партию, счёт побед и т.п. Здесь учитывались:

- правильность алгоритма,
- оптимальность решения,
- использование циклов.

#### **Наблюдаемые результаты:**

Геймифицированная структура Code.org заметно повысила активность ребят:

- большинство стремилось пройти уровни быстрее и с меньшим количеством блоков;
- участники охотно помогали друг другу находить более оптимальные решения;
- дети, которые обычно боятся ошибаться, многократно заново проходили уровни и сами обнаруживали ошибки в алгоритмах.

После занятия было проведено небольшое анкетирование по пяти ключевым параметрам (диаграмма 1), результаты представлены ниже (таблица 1).

Диаграмма 1. Результаты занятия

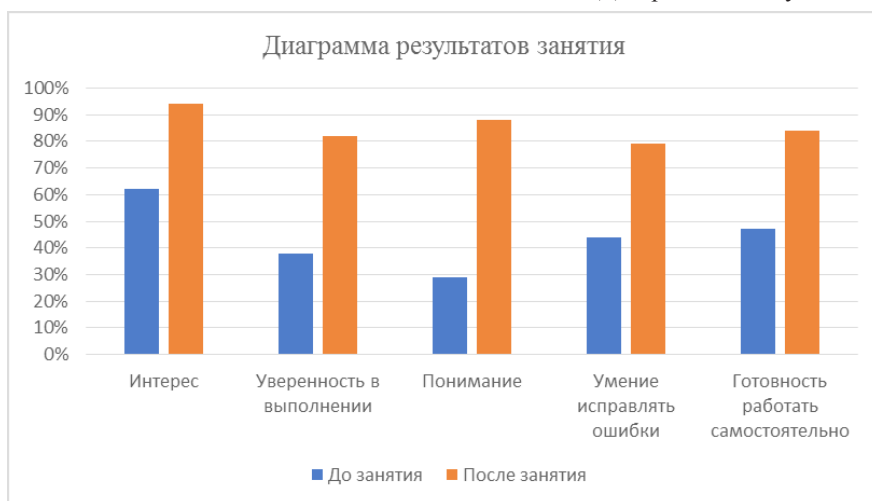


Таблица 1. Результаты занятия

Показатель	До занятия	После занятия
Интерес	62%	94%
Уверенность в выполнении	38%	82%
Понимание	29%	88%
Умение исправлять ошибки	44%	79%
Готовность работать самостоятельно	47%	84%

Создание игры на Python стало эффективным примером того, как геймификация повышает вовлечённость и формирует интерес к программированию. Учащиеся ощутили себя настоящими разработчиками, увидели результат своей работы и сделали первые шаги в сторону самостоятельного проектирования программ.

### Заключение

Прием геймификации показал высокую эффективность в обучении программированию. Она помогает превратить сложный учебный материал в увлекательный и осмысленный процесс, где каждый ученик действует активно, видит результаты своей работы, получает удовольствие от изучения, формирует важные цифровые и когнитивные навыки. Для педагога геймификация становится средством повышения успешности обучения и формирования положительного отношения к предмету. Таким образом, геймификация не заменяет традиционный урок, но усиливает его и помогает реализовать индивидуальные образовательные траектории.

### Литература и источники

1. Белкин Ф. А. Геймификация в образовании / Ф. А. Белкин. – Текст: непосредственный // Современная зарубежная психология. – 2016. – Т. 5. – № 3. – С. 28–34.
2. Варенина Л. П. Геймификация в образовании / Л. П. Варенина. – Текст: непосредственный // Историческая и социально-образовательная мысль. – 2014. – № 6-2. – С. 314–317.

3. Котенко В. В. Развитие «мягких» навыков обучающихся при изучении программирования / В. В. Котенко, А. В. Котенко. – Текст: непосредственный // Мир науки, культуры и образования. – 2023. – № 6 (103). – С. 274-278.

### *References*

1. Belkin F. A. Gejmifikaciya v obrazovanii / F. A. Belkin. – Tekst: neposredstvennyj // Sovremennaya zarubezhnaya psihologiya. – 2016. – Т. 5. – № 3. – С. 28–34.

2. Varenina L. P. Gejmifikaciya v obrazovanii / L. P. Varenina. – Tekst: neposredstvennyj // Istoricheskaya i social'no-obrazovatel'naya mysl'. – 2014. – № 6-2. – С. 314–317.

3. Kotenko V. V. Razvitie «myagkih» navykov obuchayushchihsya pri izuchenii programmirovaniya / V. V. Kotenko, A. V. Kotenko. – Tekst: neposredstvennyj // Mir nauki, kul'tury i obrazovaniya. – 2023. – № 6 (103). – С. 274-278.